## REMARKS

In an Office Action dated December 27, 2005, the Examiner objected to the specification; objected to an informality in claim 12; rejected claims 1, 12 and 14 under 35 U.S.C. 112, second paragraph, as indefinite; rejected claims 1, 5, 9-12 and 17-19 under 35 U.S.C. 103(a) as unpatentable over Doing et al. (US 5,161,166) in view of Kedem et al. (US 6,134,643), Schumann et al. (US 6,012,106), and Tanenbaum & Woodhull, "Operating Systems" (hereinafter "Tanenbaum"); and rejected claims 2-4, 13, 20 and 21 under 35 U.S.C. 103(a) as unpatentable over *Doing*, *Kedem*, *Schumann*, and *Tanenbaum*, further in view of Patterson & Hennessy, "Computer Architecture: A Quantitative Approach" and Chauvel (US 6,957,315 B2). Claims 6-8, 15 and 16 were objected to as being dependent upon a rejected base claim, but otherwise indicated to contain allowable subject matter. Claim 14 was also indicated to contain allowable subject matter, provided the indefiniteness rejection were overcome and it is re-written to include all limitations of the base claim.

***Objections to the Specification***

Two paragraphs of the specification have been amended to include the serial number of a related application. The reference to the assignee's docket number has been deleted as superfluous.

***Indefiniteness & Formal Objections***

Applicant has amended claim 1 to recite that the cached location is "more accessible to said processor than said memory is accessible to said processor..." It is believed that this amendment overcomes the Examiner's rejection.

Applicant has amended claim 12 to change the word "execution" to "executing", and to correct lack of antecedent basis.

Claim 14 has been amended to delete the phrase "automatically maintainable". This phrase appears to be confusing in scope, and does not appear to contribute significantly to patentability of the claim (applicant notes that the phrase is absent from similar claim 6).

***Prior Art***

Claim 14 has been re-written in independent form, incorporating all the limitations of original claim 12 (with amendments as explained above). As amended, claim 14 is accordingly allowable. Claim 15 is allowable as dependent on claim 14.

Claims 12 and 19 have been amended to recite that each sub-unit of a segment contains an integral number of memory pages. As applicant understands the Examiner's rejection of original claims 12 and 19, the prior art references disclose, inter alia, pre-fetch data for *cache lines*. Original claims 12 and19 made no limitation on the size of the recited "sub-units", except that they be smaller than the segment itself. As such, the "sub-units" of original claims 12 or 19 could be the size of a cache line. The Examiner indicated that original dependent claims 8 and 16, which limit the size of the sub-unit to a page, thus avoiding the interpretation that they might be cache lines, were allowable. By similar reasoning, applicant has amended claims 12 and 19 to specify that a "sub-unit" is an integral number of pages. I.e., a "sub-unit" as recited in amended claims 12 and 19 might be the size of a single page, or it might be multiple pages, but it can not be some fraction of a page, such as a cache line. Amended claims 12 and 19 are accordingly allowable.

Claim 1 has been amended to further clarify the meaning of a segment. Specifically, amended claim 1 recites that the system contains both a segment table and a page table, a segment

containing some integral number of pages, which for at least some segments is greater than one. The purpose of this amendment is to clarify that segments and pages are separate co-existing entities, segments being generally larger (although in at least some segment architectures, variable-sized segments are used in which some of the segments could be as small as a single page). As amended, claim 1 is patentable over the cited art.

A significant feature of the invention claimed in claim 1 is the use of ***data from a segment table*** for making pre-fetching selections. Various conventional techniques exist for pre-fetching data to cache. In general, these techniques involve maintaining data in the cache itself or an associated structure to indicate cache lines to be pre-fetched. Since data in cache tends to have a relatively short lifespan, these pre-fetch techniques are based on relatively short-term event sequences. Applicant observed that under certain circumstances it may be useful to base pre-fetching decisions on relatively longer term event sequences. In particular, where address translation mechanisms such as an ERAT exist, it may be useful to pre-fetch data to the ERAT based on a memory reference within the same segment. E.g., where a memory reference has been made to a segment, it can typically be expected that other pages within the same segment will be hit in the near future. By obtaining address translation data for these other pages in advance, considerable time can be saved if a subsequent reference is made to data in those pages. It is not necessary to load actual lines of referenced data (as opposed to address translation data) into cache, although it is possible to do that also.

Entries in a segment table are normally relatively long-lived. Therefore historical data used for pre-fetching decisions can be based on longer term events. It is even possible to store the segment table data to storage when none of the pages of a segment are in use, and to use this data again as required. Because longer term trends are involved, in general it is expected that granularity of pre-fetching using segment table data will be larger than that using cached data, i.e.

will be larger that individual cache lines. However, claim 1 does not limit granularity to any particular size (except that it must be smaller than the segment itself).

Applicant's amended claim 1 recites:

1. A digital data processing device, comprising:

a memory divisible into a plurality of pages, said memory containing a *segment table and a page table separate from said segment table*, said segment table having a plurality of segment table entries corresponding to respective segments in a first address space, each said *segment containing a respective integral number of said pages of said memory*, wherein, for at least some said segments, the respective integral number is greater than one, each *segment table entry containing pre-fetch data identifying a plurality of sub-units of the corresponding segment as pre-fetch candidates*;

at least one processor;

at least one structure for temporarily storing data from said memory in a location more accessible to said processor than said memory is accessible to said processor; and

a pre-fetch engine, said pre-fetch engine performing at least one pre-fetch action, said at least one pre-fetch action comprising pre-fetching selective data with respect to a sub-unit of said plurality of sub-units of a segment from said memory into said at least one structure for temporarily storing data, said *pre-fetch engine selecting said selective data for pre-fetching using said pre-fetch data in said segment table*. [emphasis added]

*Kedem* discloses a pre-fetch engine within the cache which contains a "prediction table cache". The prediction table cache is essentially a cache for some relatively small number of memory pages, and contains, for each page entry in the prediction table cache, the most recent N cache lines which were referenced. While *Kedem*'s "prediction table cache" does indeed identify a plurality of "sub-units" (cache lines) of a segment as pre-fetch candidates, it is not contained in the segment table or in any way associated with segment table data. *Kedem*'s "prediction table cache" contains entries for *memory pages* (which are *not* the same as segments), and identifies certain cache lines of selective memory pages as pre-fetch candidates..

*Schumann* discloses a system which maintains data for predicting the size of direct memory access (DMA) data transfers between memory and an I/O device. *Schumann*'s data is maintained

in a DMA controller within a host bridge. This data is maintained on a page basis, i.e., for each memory page, a corresponding entry determines the length of the predicted data transfer, so that data is fetched from memory ahead of the actual request. While *Schumann* discloses a predictive system, *Schumann*'s predictive data is not contained in a segment table entry or in any way associated with memory segments. *Schumann*'s data does not reflect a probability of a future memory reference from a processor, nor is it used for pre-fetching cache lines.

*Tanenbaum* discloses the use of memory segmentation, but does not discuss techniques for pre-fetching data to a cache structure. *Doing* discloses various basic architectural features of cache memory and memory accessing structures. The Examiner apparently concedes that it does not disclose a pre-fetch engine or the use of persistent reference history data.

The Examiner's rejection appears to be based on the argument that *Tanenbaum* teaches the equivalence of segmentation and paging, and it would therefore be obvious to base pre-fetching decisions on segment table data.

There are several responses to this argument. In the first place, applicant does not agree that *Tanenbaum* teaches the equivalence of segmentation and paging. On the contrary, *Tanenbaum* teaches that segmentation and paging are similar in certain respects and dissimilar in others. From this, it can not be said that there is any suggestion in *Tanenbaum* that a technique or device which would be applicable for use with one would necessarily be applicable for use with the other. At best, it can be said that sometimes this would be true and sometimes it would not. There are numerous constructs applicable to pages which are not applicable to segments. The mere fact that Tanenbaum mentions both paging and segmentation, and makes certain comparisons between the two, is not a suggestion that pre-fetching techniques which use page related data could be replaced by techniques which equivalently use segment related data.

Secondly, as *Tanenbaum* discloses, segmentation and paging frequently co-exist in the same system. It is just such a system which is claimed in amended claim 1. I.e., as recited in amended claim 1, segmentation does not *replace* paging, but is used in addition to or in conjunction with it. At best, the prior art suggests pre-fetching using page data.

Finally, in this rejection the Examiner is abstracting the invention and the prior art to a high-level concept. Applicant's claim 1 recites that data ***in the segment table*** is used for pre-fetching decisions. Even if one assumes that segments and pages are equivalent, an assumption that applicant does not concede and is not taught by *Tanenbaum*, none of the prior art references show data ***in a page table*** used for pre-fetching decisions. *Kedem* shows pre-fetch data in a special-purpose cache structure having entries corresponding to a selective subset of pages, but this is not a page table. *Schumann* shows pre-fetch data in a DMA controller, which again is not a page table. The Examiner has essentially abstracted applicant's claim to mean "segment related data" rather than data in the segment table entries of a segment table, and then taken the additional step of finding "segment" equivalent to "page". Thus, he reads essential limitations out of the claim.

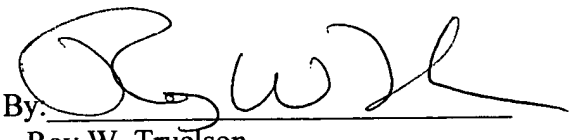
There is no equivalence of segments and pages, and there is nothing in the prior art which teaches or suggests maintaining pre-fetch data in a page table, much less maintaining pre-fetch data in the segment table entries of a segment table. For the above reasons, claim 1 as amended is not obvious over *Kedem*, *Schumann*, *Tanenbaum* and *Doing*. Claims 2-11 are dependent on claim 1 and patentable for the same reasons.

In view of the foregoing, applicant submits that the claims are now in condition for allowance, and respectfully requests reconsideration and allowance of all claims. In addition, the

Examiner is encouraged to contact applicant's attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

DAVID A. LUICK

By: 

Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725